

SISTEMI OPERATIVI

ESERCIZIO N. 1 del 19 FEBBRAIO 2001

In un **centro immersioni** viene offerta la possibilità di fare immersioni guidate. Ogni immersione deve essere accompagnata dalla **guida**, e può essere effettuata quando ci sono almeno **MIN turisti**, ma con non più di **MAX turisti** per la limitata capienza del gommone. I turisti arrivano a **gruppi** (di consistenza non superiore a MAX); i gruppi che hanno già **prenotato** hanno la precedenza su quelli che non hanno prenotato.

Si implementi una soluzione usando il costrutto monitor per modellare il **centro immersioni** e i processi per modellare i **gruppi di turisti** e la **guida** e si descriva la sincronizzazione tra i processi. Nella soluzione si massimizzi l'utilizzo delle risorse. Si discuta se la soluzione proposta può presentare starvation e in caso positivo per quali processi, e si propongano modifiche e/o aggiunte per evitare starvation.

program **vacanze**

```
const    MIN = ...; { numero minimo di turisti }
const    MAX = ...; { numero massimo di turisti }
type     tipo = (prenotato, non_pre); { tipo di turista }
type     numero = 1..MAX; { dimensione del gruppo }
```

```
type turista = process (t: tipo, n: numero)
begin
    repeat
        villaggio.entra(t, n);
        < effettua l'immersione >
        villaggio.esci(n);
    until false
end
```

```
type guida = process
begin
    repeat
        villaggio.inizio_imm;
        < guida l'immersione >
        villaggio.fine_imm;
    until false
end
```

```
type villaggio_turistico = monitor
```

```
{ variabili del monitor }
var  guida_libera : boolean;
    { stato della guida }
    coda_guida : condition;
    { coda su cui sospendere la guida }
    n_turisti : integer;
    { numero di turisti }
```

```

coda : array[tipo] of condition;
{ code su cui sospendere in attesa della guida }
in_coda : array[tipo] of integer;
{ numero di turisti sospesi per tipo }
pronti : condition;
{ turisti in attesa di iniziare l'immersione }
fine_immersione : condition;
{ turisti in attesa di finire l'immersione }

```

procedure entry **entra** (t: tipo, n: numero)

begin

```

    while guida_libera = false { se la guida non è presente }
      or n_turisti + n > MAX { o il gruppo è troppo numeroso }

```

}

```

    { o ci sono dei turisti prenotati in coda }
    or (t = non_pre and coda[prenotati].queue) do

```

begin

```

        in_coda[t] ++;
        coda[t].wait; { sospensione }
        in_coda[t] --;

```

end;

```

n_turisti := n_turisti + n ;

```

```

{ se abbiamo raggiunto il numero minimo }

```

```

if n_turisti >= MIN then

```

```

    { sveglio la guida }

```

```

    coda_guida.signal;

```

```

else

```

```

    { attendo che inizi l'immersione }

```

```

    pronti.wait;

```

end

```

procedure entry esci (n: numero)
begin
  fine_immersione.wait; { aspetta la fine dell'immersione }
  n_turisti := n_turisti - n ; { immersione finita }
end

```

```

procedure entry inizio_imm
var s: integer;
begin
  guida_libera := true;
  { sveglio i turisti prenotati in attesa di entrare }
  s := in_coda[prenotati];
  for i := 1 to s do
    coda[prenotati].signal;
  { se i prenotati in coda sono entrati tutti }
  if not coda[prenotati].queue then
    { sveglio anche i turisti non prenotati }
    begin
      s := in_coda[non_pre];
      for i := 1 to s do
        coda[non_pre].signal;
      end
    { se non c'è il numero minimo di turisti }
    if n_turisti < MIN
      coda_guida.wait; { la guida si sospende }
    { sveglio tutti i turisti pronti per l'immersione }
    while pronti.queue do
      pronti.signal;
    guida_libera := false;
  end
end

```

```

procedure entry fine_imm
begin
    { sveglia tutti i turisti in attesa di uscire }
    while fine_immersione.queue do
        fine_immersione.signal;
    end

begin { inizializzazione delle variabili }
    n_turisti := 0;
    guida_libera := false;
end

var villaggio: villaggio_turistico; { il nostro monitor }
    sp1, sp2, ... : turista (prenotato, k);
    sn1, sn2, ... : turista (non_pre, j);
    g: guida;

begin end.

```

Starvation

La soluzione proposta presenta starvation nel caso in cui i turisti non prenotati vengano sempre superati da quelli prenotati.

Si può risolvere introducendo un contatore che, dopo un numero prefissato di gruppi prenotati, dia la precedenza ai gruppi non prenotati.